UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/929,147	08/14/2001	Andreas H. Kuehnel	2007.015900	8938

7590 10/09/2003

Jeffrey A. Pyle
Williams, Morgan & Amerson, P.C.
7676 Hillmont, Suite 250
Houston, TX 77040

EXAMINER	
TRUONG, CAM Y T	
ART UNIT	**PAPER NUMBER**
2172	2

DATE MAILED: 10/09/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PRG

Office Action Summary	Application No.	Applicant(s)	
	09/929,147	KUEHNEL, ANDREAS H.	
	Examiner	Art Unit	
	Cam-Y T Truong	2172	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____ .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-56 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-7,10-12,15-18,20,22-25,27,29-32,34,36-39,41,43-46,48,50-53 and 55 is/are rejected.

7)☒ Claim(s) 8,9,13,14,19,21,26,28,33,35,40,42,47,49,54 and 56 is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-56 are pending in this Office Action.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 3, 5, 7, 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Macon, Jr. et al (or hereinafter "Macon") (USP 5715455) in view of Lawrence et al (or hereinafter "Lawrence") (USP 6253300).

As to claim 1, Macon teaches the claimed limitations:

"a plurality of containers" as the root directory is the root of all files /subdirectories (col. 4, line 55-56);

"a plurality of clusters populating each container" as the remainder of the volume after the root directory is known as the files area, which may be viewed as pools of clusters. A file B uses cluster 6, 7 and 8. A file A uses clusters 3, 4 and 5 (col. 4, lines 34-36; col.6, lines 42-47), "each cluster comprising a plurality of objects" as each cluster contains one or more logical sectors as (col. 4, lines 38-39);

"and a second data structure indicating the state of the clusters" as each cluster has a corresponding entry in the FAT that describes its current use: available, reserved, assigned to a file or unusable. For example, 0x0000 signifies an available cluster and

0xFFFF signifies an end-of cluster chain. FAT is represented as a second data structure (col. 4, lines 39-42).

Macon does not explicitly teach the claimed limitation "a first data structure indicating a state of the objects". Macon teaches that the root directory is known as the files area, which may be viewed as pools of clusters. Each cluster contains one or more sectors (col. 4, lines 34-40). Lawrence teaches other allocation map indicating which sectors or clusters are being used. This information shows that the other allocation map indicates a state of sectors as being used. The other allocation map is represented as a first data structure. Sectors are presented as objects. Being used is represented as a state of sectors (col. 12, lines 53-55).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Lawrence's teaching of other allocation map indicating which sectors or clusters are being used to Macon's system in order to track or allocate data in memory correctly and reduce number of access operations necessary to store data.

As to claim 3, Macon teaches the claimed limitation "wherein the container comprises a file" as the root directory contains files (col. 3, lines 30-35).

As to claim 5, Macon teaches the claimed limitation "the objects comprise slots in the file" as each sector having a plurality of storage locations (col. 3, lines 18-19).

As to claim 7, Macon teaches the claimed limitation "wherein at least one of the first and second data structures comprises a bitmap" as allocate bitmap for unit into temporary storage (figs. 3-4).

As to claim 10, Macon teaches the claimed limitations:

"a plurality of files" as the root directory is the root of all files /subdirectories (col. 4, line 55-56);

"a plurality of clusters populating each file," as the remainder of the volume after the root directory is known as the files area, which may be viewed as pools of clusters. A file B uses cluster 6, 7 and 8. A file A uses clusters 3, 4 and 5 (col. 4, lines 34-36; col.6, lines 42-47) "each cluster comprising a plurality of slots" as each cluster contains one or more logical sectors as (col. 4, lines 38-39);

"a directory bitmap indicating the state of the clusters" as in fig. 6 shows the process begins at 610, where various parameters are initialized, including: MAXCLUSTERS being set to the number of clusters in a particular FAT storage unit; COUNT is set to zero; pCurrentRec is defined as a pointer to the first record in the FAT storage unit; pBitmap is defined as a pointer to a bitmap for a given FAT storage unit; Current Record Index is set to zero; and Current Bitmap Index is set to zero. At decision block 615, the value of COUNT is compared to MAXCLUSTERS. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the

bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 25-47).

Macon does not explicitly teach the claimed limitation "an allocation bitmap indicating a state of the slots". Macon teaches defining Pbitmap as a pointer to a bitmap for a given FAT storage unit (col. 8, lines 33-35). Lawrence teaches bitmap indicates which sectors or clusters are being used. This information shows that the bitmap indicates a state of sectors as being used. This bitmap is represented as an allocation bitmap. Sectors are presented as slots. Being used is represented as a state of slots (col. 12, lines 53-55).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Lawrence's teaching of bitmap indicating which sectors or clusters are being used to Macon's system in order to track or allocate data in memory correctly and reduce number of access operations necessary to store data.

4. Claims 15-18, 20, 22-25, 27, 29-32, 34, 36-39, 41, 43-46, 48, 50-53, 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Macon, Jr. et al (or hereinafter "Macon") (USP 5715455).

As to claims 15, 22 and 29, Macon teaches the claimed limitations:

"tracking a state for cluster of the memory like objects in a second data structure" in fig. 6 shows the process begins at 610, where various parameters are initialized, including: MAXCLUSTERS being set to the number of clusters in a particular FAT storage unit; COUNT is set to zero; pCurrentRec is defined as a pointer to the first record in the FAT storage unit; pBitmap is defined as a pointer to a bitmap for a given FAT storage unit; Current Record Index is set to zero; and Current Bitmap Index is set to zero. At decision block 615, the value of COUNT is compared to MAXCLUSTERS. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 25-47), "and consulting at least one of the first and second data structures to manage the objects" as the file allocation tables are followed by the volume files. The boot sector contains the number of sectors per fat. This information shows that the system consults more than

one fat to manage the sectors. The first fat is represented as first data structure and the second fat is represented as second data structure (col. 4, lines 10-20).

Macon does not explicitly teach the claimed limitation " tracking a state for each of a plurality of objects populating a container in a first data structure". However, Macon teaches that the remainder of the volume after the root directory is known as the files area, which may be viewed as pools of clusters, each containing one or more logical sectors. In fig. 6 shows the process begins at 610, where various parameters are initialized, including: MAXCLUSTERS being set to the number of clusters in a particular FAT storage unit; COUNT is set to zero; pCurrentRec is defined as a pointer to the first record in the FAT storage unit; pBitmap is defined as a pointer to a bitmap for a given FAT storage unit; Current Record Index is set to zero; and Current Bitmap Index is set to zero. At decision block 615, the value of COUNT is compared to MAXCLUSTERS. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one. Since clusters contain one or more logical sectors, thus, when the system

tracks the state of clusters to determine them as being free or bad, the system should track the state of sectors of clusters too. Being free or bad is presented as a state for clusters or sectors. Sectors are represented as objects (col. 4, lines 37-40; col. 8, lines 25-47).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Macon's teaching of each containing one or more logical sectors and tracking clusters in a FAT storage unit in order to indicate status of sectors or clusters in the FAT as being either available, reserved, assigned to a file or unusable.

As to claims 16, 23 and 30, Macon teaches the claimed limitations:

"constructing the first data structure" as FAT file system (fig. 6);

"constructing the second data structure" as directory (col. 4, lines 55-56).

As to claims 17, 24, and 31, Macon teaches the claimed limitation "tracking a bitmap" as a request to read a FAT storage unit can be replaced by an unpack function which converts the FAT storage unit information stored as packed records and the end-of-file bitmap into an unpacked form. Referring therefore now to FIG. 6, the explanation will now proceed to the unpacking of the FAT. The coding of steps as described into instructions suitable to control the system processor will be understood to one having ordinary skill in the art of programming. The process begins at 610, where various parameters are initialized, including: MAXCLUSTERS being set to the number of

clusters in a particular FAT storage unit; COUNT is set to zero; pCurrentRec is defined as a pointer to the first record in the FAT storage unit; pBitmap is defined as a pointer to a bitmap for a given FAT storage unit; Current Record Index is set to zero; and Current Bitmap Index is set to zero. At decision block 615, the value of COUNT is compared to MAXCLUSTERS. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 20-47).

As to claims 18, 25 and 32, Macon teaches the claimed limitation "consulting at least one usage counter to manage the objects" as the remainder of the volume after the root directory is known as the files area which may be viewed as pools of clusters, each containing one or more logical sectors. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of

pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 20-47; col. 4, lines 37-39). Since cluster includes one or more sectors, in case a cluster includes a sector; thus, when the system sets up a counter for cluster. It means that the system sets up a counter for sector

As to claims 20, 27 and 34, Macon teaches the claimed limitation "consulting at least one list containing information extracted from usage counters to manage the objects" as the remainder of the volume after the root directory is known as the files area which may be viewed as pools of clusters, each containing one or more logical sectors. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free

nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 20-47; col. 4, lines 37-39). Since cluster includes one or more sectors, in case a cluster includes a sector; thus, when the system sets up a counter for cluster. It means that the system sets up a counter for sector and extracts value of count to manage the sectors.

As to claims 36, 43 and 50, Macon teaches the claimed limitations:

"tracking a state for cluster of the memory like objects in a directory data structure" as the root directory is known as the files area, which may be viewed as pools of clusters, each containing one or more logical sectors. In fig. 6 shows the process begins at 610, where various parameters are initialized, including: MAXCLUSTERS being set to the number of clusters in a particular FAT storage unit; COUNT is set to zero; pCurrentRec is defined as a pointer to the first record in the FAT storage unit; pBitmap is defined as a pointer to a bitmap for a given FAT storage unit; Current Record Index is set to zero; and Current Bitmap Index is set to zero. At decision block 615, the value of COUNT is compared to MAXCLUSTERS. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value;

otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 25-47; col. 4, lines 35-40);

"and consulting at least one of the first and directory data structures to manage the slots" as the file allocation tables are followed by the volume files. The boot sector contains the number of sectors per fat. This information shows that the system consults more than one fat to manage the sectors. The first fat is represented as first data structure and the second fat is represented as second data structure (col. 4, lines 10-20).

Macon does not explicitly teach the claimed limitation "tracking a state for each of a plurality of slots populating a file in a allocation data structure". However, Macon teaches that the remainder of the volume after the root directory is known as the files area, which may be viewed as pools of clusters, each containing one or more logical sectors. In fig. 6 shows the process begins at 610, where various parameters are initialized, including: MAXCLUSTERS being set to the number of clusters in a particular FAT storage unit; COUNT is set to zero; pCurrentRec is defined as a pointer to the first record in the FAT storage unit; pBitmap is defined as a pointer to a bitmap for a given FAT storage unit; Current Record Index is set to zero; and Current Bitmap Index is set to zero. At decision block 615, the value of COUNT is compared to MAXCLUSTERS.

In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one. Since clusters contain one or more logical sectors, thus, when the system tracks the state of clusters to determine them as being free or bad, the system should track the state of sectors of clusters too. Being free or bad is presented as a state for clusters or sectors. Sectors are represented as slots (col. 4, lines 37-40; col. 8, lines 25-47).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Macon's teaching of each containing one or more logical sectors and tracking clusters in a FAT storage unit in order to indicate status of sectors or clusters in the FAT as being either available, reserved, assigned to a file or unusable.

As to claims 37, 44 and 51, Macon teaches the claimed limitation:

"constructing the allocation data structure" as FAT file system (fig. 6);

"constructing the directory data structure" as directory (col. 4, lines 55-56).

As to claims 38, 45 and 52, Macon teaches the claimed limitation "tracking in a bitmap" as " as a request to read a FAT storage unit can be replaced by an unpack function which converts the FAT storage unit information stored as packed records and the end-of-file bitmap into an unpacked form. Referring therefore now to FIG. 6, the explanation will now proceed to the unpacking of the FAT. The coding of steps as described into instructions suitable to control the system processor will be understood to one having ordinary skill in the art of programming. The process begins at 610, where various parameters are initialized, including: MAXCLUSTERS being set to the number of clusters in a particular FAT storage unit; COUNT is set to zero; pCurrentRec is defined as a pointer to the first record in the FAT storage unit; pBitmap is defined as a pointer to a bitmap for a given FAT storage unit; Current Record Index is set to zero; and Current Bitmap Index is set to zero. At decision block 615, the value of COUNT is compared to MAXCLUSTERS. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster

(0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 20-47).

As to claims 39, 46 and 53, Macon teaches the claimed limitation " consulting at least one usage counter to manage the slots" as the remainder of the volume after the root directory is known as the files area which may be viewed as pools of clusters, each containing one or more logical sectors. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 20-47; col. 4, lines 37-39). Since cluster includes one or more sectors, in case a cluster includes a sector; thus, when the system sets up a counter for cluster. It means that the system sets up a counter for sector and extracts value of count to manage the sectors.

As to claims 41, 48 and 55, Macon teaches the claimed limitation "consulting at least one list containing information extracted from usage counters to manage the slots"

as the remainder of the volume after the root directory is known as the files area which may be viewed as pools of clusters, each containing one or more logical sectors. In the event that COUNT is greater than MAXCLUSTERS, the YES branch is taken to 620 to exit the unpack function; otherwise, the NO branch is taken to serve as input to decision block 625. Decision block 625 examines whether the value of the bit at the position defined by the value of pBitmap+Current Bitmap Index is set. If set, the YES branch is taken where 630 assigns the value of the current cluster address at 0xFFFF, the end-of-file cluster value; otherwise the NO branch is taken to decision block 635. At decision block 635, the current cluster address is examined as constituting a free (0x0000) or bad cluster (0xFFF7). If the given cluster is neither free nor bad, the NO branch is taken to 640, where the Current Record Index is incremented by one (col. 8, lines 20-47; col. 4, lines 37-39). Since cluster includes one or more sectors, in case a cluster includes a sector; thus, when the system sets up a counter for cluster. It means that the system sets up a counter for sector and extracts value of count to manage the sectors.

5. Claims 2, 4, 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Macon, Jr. et al (or hereinafter "Macon") (USP 5715455) in view of Lawrence et al (or hereinafter "Lawrence") (USP 6253300) and further in view of Lehman (USP 5732402).

As to claim 2, Macon and Lawrence disclose the claimed limitation subject matter in claim 1, except the claimed limitation "a plurality of container control data". However, Lehman teaches that management of the LOB data space, including allocation of space

and storage/retrieval of data, is controlled by allocation pages. Allocation pages are represented as a plurality of container (col. 5, lines 43-45).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Lehman's teaching of management of the LOB data space, including allocation of space and storage/retrieval of data, is controlled by allocation pages to Macon and Lawrence's system in order to set flags in storage for indicating whether a space is currently occupied or is free to be used.

As to claims 4 and 11, Macon and Lawrence disclose the claimed limitation subject matter in claim 3, except the claimed limitation "wherein the file is a page file or a swap file". However, Lehman teaches that the pages in the space allocation file include a means of indicating free blocks of storage location controls the storage of data in the buddy space. Applicant shows the file is a page file or a swap file. In this case, examiner indicates the file is a page file. Thus, the space allocation file of pages is represented as a page file (col. 5, lines 50-55).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Lehman teaching of the pages in the space allocation file include a means of indicating free blocks of storage location controls the storage of data in the buddy space to Macon and Lawrence's system in order to store a large amount objects in the context of a paging memory.

6. Claims 6 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Macon, Jr. et al (or hereinafter "Macon") (USP 5715455) in view of Lawrence et al (or hereinafter "Lawrence") (USP 6253300) and further in view of Yamagami et al (or hereinafter "Yamagami") (USP 6256282).

As to claim 6, Macon discloses the claimed limitation subject matter in claim 1, except the claimed limitation "wherein each cluster comprises 16 objects". However, Yamagami teaches that one cluster is constituted by 16 sectors (col. 16, line 9).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Yamagami's teaching of one cluster is constituted by 16 sectors to Macon's system in order to store a large data object in a cluster.

As to claim 12, Macon discloses the claimed limitation subject matter in claim 1, except the claimed limitation "wherein each cluster comprises 16 slots". However, Yamagami teaches that one cluster is constituted by 16 sectors (col. 16, line 9).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Yamagami's teaching of one cluster is constituted by 16 sectors to Macon's system in order to store a large data object in a cluster.

Allowable Subject Matter

7. Claims 8-9, 13-14, 19, 21, 26, 28, 33, 35, 40, 42, 47, 49, 54 and 56 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in

independent form including all of the limitations of the base claim and any intervening claims.

As to claims 8, 13, 19, 26, 33, 40, 47 and 54, none of the available prior art of record teaches or fairly suggest "a counter of how many sets of 2 adjacent bits are set in the words of the directory bitmap; a counter of how many sets of 4 adjacent bits are set in the words of the directory bitmap; a counter of how many sets of 8 adjacent bits are set in the words of the directory bitmap; a counter of how many sets of 16 adjacent bits are set in the words of the directory bitmap; a counter of how many sets of 32 adjacent bits are set in the words of the directory bitmap; and a counter of how many sets of 64 adjacent bits are set in the words of the directory bitmap".

As to claims 21, 28, 35, 42, 49, 54 and 56, none of the available prior art of record teaches or fairly suggest "a first list containing information indicating that the directory data structure for files in this list contains clusters of at least four adjacent free bits; a second list containing information indicating that the directory data structure for files in this list is not empty, but contains no clusters of four adjacent free bits; a third list containing information indicating that the directory data structure for files in this list is empty, but allocation bitmap still shows free slots"

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Haderle et al (USP 6334134).

Contact Information

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cam-Y Truong whose telephone number is (703-605-1169). The examiner can normally be reached on Mon-Fri from 8:00AM to 4:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu, can be reached on (703-305-4393). The fax phone numbers for the organization where this application or proceeding is assigned is (703)-746-7239 (formal communications intended for entry), or: (703)-746-7240 (informal communication labeled PROPOSED or DRAFT).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703-305-3900).

Cam-Y Truong

9/23/03

SHAHID ALAM
PRIMARY EXAMINER